

### Amendments to the Specification

*Please replace Specification paragraph at page 8, lines 11-20 with the following amended paragraph:*

a<sup>1</sup>  
Assume, for example, that NIC 1 116 has an on-board encryption ASIC, but NIC 2 118 and NIC 3 120 do not. As will be discussed in more detail below, in such a circumstance, encryption for NIC 2 118 and NIC 3 120 can be supported by routing encryption requests through NIC 1 116 encryption hardware and then repackaging the resultant encrypted data for delivery to NIC 2 118 and/or NIC 3 120 by way of the virtual protocol stack 202. That is, in one embodiment, network traffic to be encrypted would go from protocol stack 100, to the intermediary 102, to the virtual driver 204, which communicates with the NIC 1 driver 104 to have NIC 1 116 perform the encryption. The encrypted data is received by the virtual driver 204, given to the virtual protocol stack 202, which then re-sends the data for transmission by NIC 2 118 or NIC 3 120.

*Please replace Specification paragraph at page 14, lines 7-14 with the following amended paragraph:*

a<sup>2</sup>  
If, however, non-regular traffic is received, e.g., secondary use data packets, then these packets are delivered to the backup network interface members such that they are balanced 372 across all available unused team members. If 370 the primary network interface has available resources, however, to process encryption tasks, then the primary adapter interleaves 374 secondary task processing with its primary transmission and receipt of network traffic. Remaining task processing is balanced 372 across all available unused team members. It is expected that appropriate queuing strategies will be employed to keep all adapters busy.

*Please replace Specification paragraph at page 15, lines 4-15 with the following amended paragraph:*

a<sup>3</sup>  
FIG. 7 and the following discussion are intended to provide a brief, general description of a suitable computing environment in which portions of the invention may be implemented. An

a3  
exemplary system for implementing the invention includes a computing device 400 having system bus 402 for coupling together various components within the computing device. The system bus may be any of several types of bus structures, such as PCI, AGP, VESA, etc. Typically, attached to the bus 402 are processors 404 such as Intel Pentium® processors, programmable gate arrays, etc., a memory 406 (e.g., RAM, ROM, NVRAM), computing-device readable storage-media 408, a video interface 410, input/output interface ports 412, and a network interface. A modem 414 may provide an input and/or output data pathway, such as for user input/output, and may operate as a network interface in lieu of or in conjunction with other network interfaces 416.

*Please replace Specification paragraph at page 15 line 22 – page 16 line 4 with the following amended paragraph:*

a4  
The exemplary computing device 400 can store and execute a number of program modules within the memory 406, and computing-device readable storage-media 408. The executable instructions may be presented in terms of algorithms and/or symbolic representations of operations on data bits within a computer memory, as such representation is commonly used by those skilled in data processing arts to most effectively convey the substance of their work to others skilled in the art. Here, and generally, an algorithm is conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities, and can take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. Appropriate physical quantities of these signals are commonly referred to as bits, values, elements, symbols, characters, terms, numbers, or the like.